

Distributed Average Consensus in Digraphs

Christoforos N. Hadjicostis and Alejandro D. Domínguez-García

Abstract In this chapter, we address the average-consensus problem for a distributed system whose components (nodes) can exchange information via interconnections (links) that form an arbitrary, possibly directed, communication topology (digraph). Specifically, we discuss how the nodes can asymptotically obtain the average of their initial values by describing two different types of algorithms, one based on weight adaptation and one based on ratio consensus.

1 Motivation

The design of protocols and algorithms for distributed computation, and control/decision-tasks has attracted significant attention by the computer science, communication, and control communities (e.g., [15, 14, 16]). Given a set of interconnected nodes (which could be sensors in a sensor network, routers in a communication network, or unmanned vehicles in a multi-agent system), motivational applications range from (i) averaging individual measurements (e.g., when each node provides a local measurement of a global quantity); (ii) transmitting data from one or multiple sources to one or multiple sinks; (iii) coordinating node speed or direction; or (iv) electing a leader.

In a typical consensus problem, each node possesses an initial value and the nodes need to follow a strategy to distributively calculate the same function of these initial values. The consensus problem has received extensive attention from the computer science community (see [15] for an introduction) and the control community

C. N. Hadjicostis is with the Department of Electrical and Computer Engineering at the University of Cyprus, Nicosia, Cyprus, and also with the Department of Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA. E-mail: chadjic@UCY.AC.CY. A. D. Domínguez-García is with the Department of Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA. E-mail: aledan@ILLINOIS.EDU.

[20, 16], due to its applicability to topics such as cooperative control, multi-agent systems, and modeling of flocking behavior in biological and physical systems (see, e.g., [12, 19, 21] and references therein). When the value to which the nodes agree is the average of the initial values we say that the nodes reach *average consensus*. *Asymptotic average consensus* is reached if (typically, after executing an iterative strategy) the nodes asymptotically converge to the average of their initial values. There are a number of different characteristics that are important in average-consensus problems, including distributivity constraints (e.g., the need for each node to rely solely on locally available information), time-varying interconnections, computational and communication complexity, convergence to the average or close to it (in case the algorithm is asymptotic), speed of convergence, and others.

This chapter studies on the average-consensus problem when the interconnection topology is described by a fixed *directed graph* (digraph) and focuses primarily on establishing convergence; more details about speed of convergence and computational/communication complexity (as well as pertinent examples) can be found in the references provided. The presence of an asymmetric information structure makes the average-consensus task in digraphs particularly challenging due to the fact that nodes cannot (easily) provide acknowledgements or more generally inform the nodes that guide their decisions about the actions they are taking. Asymmetric topologies can arise in a variety of realistic scenarios (e.g., if nodes transmit at different power strengths or if interference levels are not uniform at each node).

2 Problem Statement

The exchange of information between components (nodes) of a distributed system can be conveniently described by a directed graph (digraph) $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{1, 2, \dots, n\}$ is the vertex set (each vertex corresponds to a component/node), and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of directed edges, where $(j, i) \in \mathcal{E}$ if node j can receive information from node i . By convention, we assume no self-loops in \mathcal{G} (i.e., $(j, j) \notin \mathcal{E}$ for all $j \in \mathcal{V}$). The graph is undirected if and only if whenever $(j, i) \in \mathcal{E}$, then also $(i, j) \in \mathcal{E}$, i.e., if node j can receive information from node i , then node i can also receive information from node j . All nodes that can transmit information to node j are said to be in-neighbors of node j and are represented by the set $\mathcal{N}_j^- = \{i \in \mathcal{V} \mid (j, i) \in \mathcal{E}\}$. The number of in-neighbors of j is called the in-degree of j and is denoted by \mathcal{D}_j^- (i.e., $\mathcal{D}_j^- = |\mathcal{N}_j^-|$). Nodes that receive information from node j are said to be the out-neighbors of node j , and are represented by the set $\mathcal{N}_j^+ = \{l \in \mathcal{V} \mid (l, j) \in \mathcal{E}\}$. The number of out-neighbors of j is called the out-degree of j and is denoted by \mathcal{D}_j^+ (i.e., $\mathcal{D}_j^+ = |\mathcal{N}_j^+|$).

Let V_j be the initial value of node j . In the average-consensus problem, the objective is to have all the nodes calculate the average of these initial values, which we denote by $\mu = \frac{\sum_{\ell=1}^n V_\ell}{n}$. Depending on the assumptions, nodes may or may not know n , and they will presumably require several rounds of message exchanges in order to obtain μ (perhaps obtaining the values of V_ℓ , $\ell = 1, 2, \dots, n$, in the process). In the

algorithms we present in this chapter, in order to obtain μ , each node j maintains some value $\pi_j[k]$ at round k , and performs a linear iteration of the form

$$\pi_j[k] = p_{jj}[k]\pi_j[k-1] + \sum_{i \in \mathcal{N}_j^-} p_{ji}[k]\pi_i[k-1]. \quad (1)$$

In other words, each node j updates its value to be a linear combination of its own previous value and the values of its in-neighbors using its own self-weight ($p_{jj}[k]$) and the weights ($p_{ji}[k]$, $i \in \mathcal{N}_j^-$) on its incoming links. If we let $\pi[k] = [\pi_1[k], \pi_2[k], \dots, \pi_j[k], \dots, \pi_n[k]]'$ (where $'$ denotes vector transposition), then for analysis purposes (1) can be written in matrix form as

$$\pi[k] = P[k]\pi[k-1], \quad (2)$$

where the weight matrix $P[k] = [p_{ji}[k]]$ (with $p_{ji}[k]$ as the entry at the j th row- i th column of matrix $P[k]$).

In (1), the $p_{ji}[k]$'s are a set of (time-varying) weights that need to be chosen (along with the initial conditions $\pi[0]$) so that all $\pi_j[k]$ converge for large k to μ . Node j can only choose its self-weight and the weights on its out-going links, i.e., node j can choose values for $\{p_{lj}[k] \mid l = 1, 2, \dots, n\}$, with the constraint that $p_{lj}[k] = 0$ for all l such that $l \notin \mathcal{N}_j^+$. It is assumed that each node can observe but cannot control the (likely different) values on each of its incoming links, and cannot necessarily identify the sender node associated with each value. These assumptions hold naturally for most interconnection topologies that form a digraph (in fact, in many practical situations additional information may be available at each node).

Remark 1. For the case when the weights $p_{lj}[k]$'s are fixed for all $k \geq 0$ (i.e., $p_{lj}[0] = p_{lj}[1] = \dots = p_{lj}[k] = \dots := p_{lj}$ for $(l, j) \in \mathcal{E}$), as stated in [19, 21] in various forms, the necessary and sufficient conditions for the iteration in (2) (with $P[k] = P$, $P(l, j) = p_{lj}$, and $\pi[0] = [V_1, V_2, \dots, V_j, \dots, V_n]'$) to asymptotically reach average consensus are: (i) P has a simple eigenvalue at 1, with left eigenvector $[1, 1, 1, \dots, 1]$ and right eigenvector $[1, 1, 1, \dots, 1]'$, and (ii) all other eigenvalues of P have magnitude strictly less than 1. If one focuses on nonnegative weights, these conditions are equivalent to the weight matrix P being a primitive doubly stochastic matrix. This is easily achievable in an undirected graph: assuming nodes know the total number of nodes n or an upper bound $n' \geq n$, each node j can easily choose fixed (nonnegative) weights on its out-going links so that $\sum_l p_{lj} = \sum_i p_{ji} = 1, \forall j$, by setting $p_{jj} = 1 - \frac{\mathcal{D}_j}{n'}$, $p_{lj} = \frac{1}{n'}$ if $(l, j) \in \mathcal{E}$, and $p_{lj} = 0$ if $(l, j) \notin \mathcal{E}$, where $\mathcal{D}_j = \mathcal{D}_j^+ = \mathcal{D}_j^-$. As long as the undirected graph is connected, this choice results in a primitive doubly stochastic (symmetric) weight matrix P . Another simple choice that results in a primitive doubly stochastic (symmetric) weight matrix P in connected undirected graphs are the *Metropolis* weights in [22] where $p_{lj} = \frac{1}{1 + \max(\mathcal{D}_l, \mathcal{D}_j)}$ if $(l, j) \in \mathcal{E}$, $p_{lj} = 0$ if $(l, j) \notin \mathcal{E}$, and $p_{jj} = 1 - \sum_i p_{ji}$. In a digraph, however, a given node j may not necessarily have $\mathcal{D}_j^+ = \mathcal{D}_j^-$, and it is not as straightforward for nodes to determine appropriate weights so that $\sum_l p_{lj} = \sum_i p_{ji} = 1, \forall j$. [Strategies to obtain matrices with appropriately scaled rows and columns (including doubly stochastic matrices

as a special case) have been studied under the umbrella of *matrix scaling* (see, for example, [17]) *without*, however, paying attention to distributivity constraints.]

3 Distributed Strategies for Average Consensus in Digraphs

In this section we describe two strategies that can be used by nodes in a given strongly connected digraph to distributively reach average consensus. The first method relies on an embedded distributed weight adjustment algorithm, whereas the second method relies on a ratio consensus approach that simultaneously runs two (coupled) iterations, and has each node take the ratio of its two iteration values. Both algorithms require (at the base level) that each node knows its out-degree; this requirement is rather mild, as in most protocols for ad-hoc network discovery each node not only knows which nodes it receives information from, but it also knows which nodes it transmits information to. We also discuss how broadcasting can be accommodated by ensuring that each node sends identical information to each of its out-neighbors.

3.1 Distributed Weight Adjustment

This section summarizes a *weight balancing* algorithm that enables the nodes in a strongly connected digraph to asymptotically reach weights that form a primitive doubly stochastic matrix. This algorithm has been presented in [6] which also showed that the weight adaptation can be combined with iteration (1) to ensure that the nodes asymptotically reach average consensus. We next describe these ideas starting from the distributed weight adjustment algorithm.

For distributed weight adjustment, each node iteratively updates its self-weight and the weights on its out-going links based on the sum of the weights on its incoming links. More specifically, at each iteration k , node j maintains a parameter $\delta_j[k]$ which determines its self-weight and the weights on its out-going links as $p_{jj}[k] = 1 - \delta_j[k]$ and $p_{lj}[k] = c_{lj}\delta_j[k]$, where c_{lj} are constants chosen by node j at initialization so that: (i) $\sum_{l,l \neq j} c_{lj} = 1$, and (ii) $c_{lj} > 0$ if node l can receive information from node j (i.e., if $(l, j) \in \mathcal{E}$), and $c_{lj} = 0$ if node l cannot receive information from node j (i.e., if $(l, j) \notin \mathcal{E}$). For notational convenience, we will take $c_{jj} = 0$ for $j = 1, 2, \dots, n$ so that $\sum_{l,l \neq j} c_{lj} = \sum_l c_{lj} = 1$. [Note that for broadcasting to be possible, node j should choose identical c_{lj} which implies that $c_{lj} = \frac{1}{\mathcal{O}_j^+}$ for $(l, j) \in \mathcal{E}$.]

Initially, node j sets $\delta_j[0] = \frac{\mathcal{O}_j^+}{1 + \mathcal{O}_j^+}$ (which implies that $p_{jj}[0] = \frac{1}{1 + \mathcal{O}_j^+}$ and $p_{lj}[0] = c_{lj} \frac{\mathcal{O}_j^+}{1 + \mathcal{O}_j^+}$). At each iteration k , $k \geq 1$, node j gathers the weights $p_{ji}[k]$, $(j, i) \in \mathcal{E}$, on its incoming edges and updates the value of $\delta[k]$ as

$$\delta_j[k] = \begin{cases} \delta_j[k-1]\rho_j[k-1], & \text{if } \rho_j[k-1] \leq 1, \\ 1 - \frac{1}{\rho_j[k-1]}(1 - \delta_j[k-1]), & \text{if } \rho_j[k-1] > 1, \end{cases} \quad (3)$$

where $\rho_j[k-1] = \sum_i p_{ji}[k-1]$. Based on $\delta_j[k]$, node j updates its self-weight and the weights on its out-going links as $p_{jj}[k] = 1 - \delta_j[k]$ and $p_{lj}[k] = c_{lj}\delta_j[k]$, $l \neq j$.

Since the initial value of $\delta_j[0]$ and the update of $\delta_j[k]$ ensure $0 \leq \delta_j[k] \leq 1$ for all k , it can be easily verified that the $n \times n$ matrix $P[k] = [p_{ji}[k]]$ (with $p_{ji}[k]$ as its $(j, i)^{th}$ entry) will be a nonnegative column-stochastic matrix (i.e., $p_{lj}[k] \geq 0$ and $\sum_{l=1}^n p_{lj}[k] = 1$, $j = 1, 2, \dots, n$). In fact, one can also check that, as long as the diagonal entries of $P[k]$ are strictly smaller than one and at least one diagonal entry is strictly positive, then $P[k]$ will be primitive. This is due to the fact that $P[k]$ corresponds to a graph that is strongly connected [18].

Note that matrix $P[k]$ can also be written as

$$P[k] = \bar{P}\Delta[k] + (I - \Delta[k]), \quad (4)$$

where I is the $n \times n$ identity matrix, $\Delta[k] = \text{diag}(\delta_1[k], \delta_2[k], \dots, \delta_n[k])$ is a diagonal matrix with entries $\delta_j[k]$ on the diagonal, and $\bar{P} = [c_{lj}]$ is the constant weight matrix chosen at initialization, i.e.,

$$\bar{P} = \begin{bmatrix} 0 & c_{12} & c_{13} & \dots & c_{1n} \\ c_{21} & 0 & c_{23} & \dots & c_{2n} \\ c_{31} & c_{32} & 0 & \dots & c_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & c_{n3} & \dots & 0 \end{bmatrix}. \quad (5)$$

The weight update described by the update of the $\delta_j[k]$'s in (3) and the parameterization in (4) can be shown to drive the weights, as k goes to infinity, to a limiting weight matrix that is doubly stochastic and primitive. In other words, the sequence of matrices $P[0], P[1], P[2], \dots, P[k], \dots$ converges to a doubly stochastic and primitive matrix P_{ss} as long as the given digraph is strongly connected [6].

The weight adaptation algorithm can be combined with iteration (1) to ensure that the nodes reach asymptotic average consensus. More specifically, for any $k \geq 1$, the nodes update their values according to

$$\pi[k] = P[k]\pi[k-1], \quad (6)$$

where $P[k]$ is determined by (3) and the parameterization in (4). Note that, at each iteration k , each node j is supposed to perform two tasks: an update of $\delta_j[k]$ (and, thus, of its self-weight and the weights on its out-going links), followed by an update of its value $\pi_j[k]$. Then, taking into account the fact that the sequence $P[0], P[1], \dots, P[k], \dots$ consists of column stochastic (and primitive) matrices, the steady-state solution of (6), with initial conditions $\pi_j[0] = V_j$, $\forall j$, denoted by π^{ss} , is such that

$$\pi_j^{ss} = \mu, \forall j = 1, 2, \dots, n. \quad (7)$$

The following theorem summarizes the discussions in this section; its proof can be found in [6].

Theorem 1. Consider a strongly connected digraph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with $\mathcal{V} = \{1, 2, \dots, n\}$ where each node $j \in \mathcal{V}$ has some initial value $\pi_j[0] = V_j$. Suppose that nodes set $\delta_j[0] = \frac{\mathcal{D}_j^+}{1 + \mathcal{D}_j^+}$, for $j \in \mathcal{V}$, and update their values, for $k \geq 1$, according to (6) where $P[k] = \bar{P}\Delta[k] + (I - \Delta[k])$, and $\Delta[k] = \text{diag}(\delta_1[k], \delta_2[k], \dots, \delta_j[k], \dots, \delta_n[k])$ with $\delta_j[k]$, $\forall j \in \mathcal{V}$, updated according to (3). Then, $\lim_{k \rightarrow \infty} P[k] = P_{ss}$ where P_{ss} is doubly stochastic and primitive, and $\lim_{k \rightarrow \infty} \pi[k] = \pi^{ss}$ where the steady-state vector π^{ss} satisfies $\pi_j^{ss} = \mu$, $\forall j = 1, 2, \dots, n$.

3.2 Ratio Consensus

This section summarizes the so called *ratio-consensus* algorithm [5], a distributed algorithm that enables the nodes of a multi-component system to reach average consensus. For gossiping-type algorithms, an equivalent approach was also proposed in [2], which is a generalization of the algorithm proposed in [13]; another recent application of ratio consensus appears in [1], where a modified distributed Kalman consensus utilizes ratio consensus to obtain an unbiased estimate for static or dynamic communication networks. However, the idea of ratio-consensus can be traced back much earlier; see the discussion on weak convergence in [18, pp. 88-89]. The ratio-consensus algorithm performs two iterative computations in parallel and allows each node to asymptotically obtain the exact average of the values the nodes possess as the ratio of the two state variables that each node maintains. The following theorem (see, for example, [5]) summarizes the basic version of ratio consensus.

Theorem 2. Consider a strongly connected digraph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with $\mathcal{V} = \{1, 2, \dots, n\}$ where each node $j \in \mathcal{V}$ has some initial value V_j . Each node j maintains, at iteration k , state variables $y_j[k]$ and $z_j[k]$ and updates them as follows:

$$y_j[k+1] = \sum_{i \in \mathcal{N}_j^- \cup \{j\}} y_i[k] / (1 + \mathcal{D}_i^+), \quad k \geq 0, \quad (8)$$

$$z_j[k+1] = \sum_{i \in \mathcal{N}_j^- \cup \{j\}} z_i[k] / (1 + \mathcal{D}_i^+), \quad k \geq 0, \quad (9)$$

where $y_j[0] = V_j$, and $z_j[0] = 1$, for all $j \in \mathcal{V}$. Let $\pi_j[k] = \frac{y_j[k]}{z_j[k]}$; then, we have $\lim_{k \rightarrow \infty} \pi_j[k] = \frac{\sum_{\ell} y_{\ell}[0]}{\sum_{\ell} z_{\ell}[0]} = \frac{\sum_{\ell} V_{\ell}}{n} = \mu$, $\forall j \in \mathcal{V}$.

Remark 2. Letting $y[k] = [y_1[k], y_2[k], \dots, y_n[k]]'$ and $z[k] = [z_1[k], z_2[k], \dots, z_n[k]]'$, we can write the iterations in the above theorem in matrix form as $y[k+1] = P_c y[k]$, and $z[k+1] = P_c z[k]$ where the initial conditions are the same as in the theorem and P_c is a matrix with entries $P_c(l, j) = \frac{1}{1 + \mathcal{D}_j^+}$ for all $l \in \mathcal{N}_j^+$ (zero otherwise).

Notice that P_c is a column stochastic matrix that is also primitive as long as the underlying digraph is strongly connected. The ratio consensus algorithm has each node j calculate at each time step k the ratio $\pi_j[k] = \frac{y_j[k]}{z_j[k]}$. As long as P_c is primitive column stochastic, it can be shown that the ratio $\pi_j[k]$ asymptotically converges to $\frac{\sum_{\ell} y_{\ell}[0]}{\sum_{\ell} z_{\ell}[0]}$. Thus, by appropriately choosing the initial conditions of ratio consensus, we can compute arbitrary weighted linear combinations of the initial values of the nodes. For example, if $y_j[0] = c_j V_j$ and $z_j[0] = c'_j$, then $\lim_{k \rightarrow \infty} \pi_j[k] = \frac{\sum_{\ell} y_{\ell}[0]}{\sum_{\ell} z_{\ell}[0]} = \frac{\sum_{\ell} c_{\ell} V_{\ell}}{\sum_{\ell} c'_{\ell}}$, $\forall j \in \mathcal{V}$. Note that, in general, none of the sequence $y_j[k]$, $z_j[k]$, $\pi_j[k]$ for $k \geq 0$ are monotone.

3.3 Other Approaches

The work in [9] proposed a broadcast-based gossip algorithm that relies on each node knowing its out-degree and performing an iteration that involves two variables (that are coupled); the authors show that average consensus is reached but a formal proof of convergence is still open. The authors of [3] use a similar approach (using so-called “surplus variables”) and prove convergence for certain small-gain values of the coupling coefficients. Techniques that rely on surplus-like compensation methods to reach average consensus in the presence of unreliable communication links have received attention relatively recently in [8, 4]. Extensions of ratio consensus to handle packet drops and delays in digraphs appear in [7, 11]. Along similar lines (but not motivated by randomization induced by packet drops), the authors of [10] consider randomized discrete-time consensus systems defined over digraphs that preserve the average “on average,” by providing an upper bound on the mean square deviation of the consensus value from the exact average.

4 Conclusion

In this chapter, we provided an introduction to the problem of average consensus in distributed control systems (with underlying possibly asymmetric communication topologies), and described two algorithms that can be used to asymptotically obtain the average.

References

1. M. Alighanbari and J. P. How. An unbiased Kalman consensus algorithm. In *American Control Conference (ACC 2006)*, 2006.

2. F. Benezit, V. Blondel, P. Thiran, J. N. Tsitsiklis, and M. Vetterli. Weighted gossip: Distributed averaging using non-doubly stochastic matrices. In *Proc. IEEE International Symposium on Information Theory*, pages 1753–1757, June 2010.
3. K. Cai and H. Ishii. Average consensus on general strongly connected digraphs. *Automatica*, 48(1):2750–2761, November 2012.
4. Y. Chen, R. Tron, A. Terzis, and R. Vidal. Corrective consensus: Converging to the exact average. In *Proc. IEEE Conf. Decision and Control*, pages 1221–1228, December 2010.
5. A. D. Domínguez-García and C. N. Hadjicostis. Coordination and control of distributed energy resources for provision of ancillary services. In *Proc. IEEE SmartGridComm*, pages 537–542, October 2010.
6. A. D. Domínguez-García and C. N. Hadjicostis. Distributed matrix scaling and application to average consensus in directed graphs. *IEEE Transactions on Automatic Control*, 58(3):667–681, March 2013.
7. A. D. Domínguez-García, C. N. Hadjicostis, and N. H. Vaidya. Resilient networked control of distributed energy resources. *IEEE Journal on Selected Areas in Communications*, 30(6):1137–1148, June 2012.
8. F. Fagnani and S. Zampieri. Average consensus with packet drop communication. *SIAM Journal on Control and Optimization*, 48(1):102–133, 2009.
9. M. Franceschelli, A. Giua, and C. Seatzu. Distributed averaging in sensor networks based on broadcast gossip algorithms. *IEEE Sensors Journal*, 11(3):808–817, March 2011.
10. P. Frasca and J. M. Hendrickx. On the mean square error of randomized averaging algorithms. *Automatica*, 49(8):2496–2501, 2013.
11. C. N. Hadjicostis and T. Charalambous. Average consensus in the presence of delays in directed graph topologies. *IEEE Transactions on Automatic Control*, 59(3):763–768, March 2014.
12. A. Jadbabaie, Jie Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, June 2003.
13. D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Proc. of the 44th Annual IEEE Symposium on Foundations of Computer Science*, FOCS 2003, pages 482–491, October 2003.
14. R. Koetter and M. Médard. An algebraic approach to network coding. *IEEE/ACM Transactions on Networking*, 11(5):782–795, October 2003.
15. N. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers, San Mateo, CA, 1996.
16. R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, January 2007.
17. B. N. Parlett and T. L. Landis. Methods for scaling to doubly stochastic form. *Linear Algebra and its Applications*, 48:53–79, 1982.
18. E. Seneta. *Non-negative Matrices and Markov Chains*. Springer, New York, NY, revised printing edition, 2006.
19. S. Sundaram and C. N. Hadjicostis. Distributed function calculation and consensus using linear iterative strategies. *IEEE Journal on Selected Areas in Communications*, 26(4):650–660, May 2008.
20. J. N. Tsitsiklis. *Problems in decentralized decision making and computation*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1984.
21. L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Systems and Control Letters*, 53(1):65–78, September 2004.
22. L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, pages 63–70, 2005.